Escola Universitària
Politècnica de Mataró

Centre adscrit a:

UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Enginyeria tècnica d'informàtica de gestió

Gestió previsió pressupostària

Memòria

Narcís Mir Gibert
PONENT: Eduard de Bru de Sala

PRIMAVERA 2011

TecnoCampus
Mataró-Maresme

Dedicatòria

Vull dedicar aquesta obra d'art al meu ratolí, un excel·lent exemplar de color negre molt suau i tendre. El qual sempre va aguantar les meves frustracions i plors fins que un trist vint de juny va passar a millor vida al ser brutalment llençat contra la paret.

Logitech, sempre et duré al més profund del meu cor.

Agraïments

Abans que ningú vull agrair al meu farmacèutic la seva cura i dedicació al proporcionar-ne tota mena d'aspirines, ibuprofenos i altres substàncies, que quedaria bastant lleig anomenar-les, que m'han fet la carrera més suportable.

Gràcies Òscar.

Finalment, agrair a Sant Joan i al folklore català, una de les millors festes en les quals em podré desfer de cinc anys de penes i enrabiades materialitzades en papers i llibres.

Gràcies Sant Joan i gràcies Catalunya

Resum

Aquest any la nostre universitat EUPMT ha canviat la seva ubicació per un altre de més ampli i adient sota el nom de Tecnocampus. Degut bàsicament, a que l'escola ha anat creixent i l'anterior edifici havia quedat antiquat i petit. De la mateixa manera que el creixement de l'escola va desembocar en el trasllat en l'actual edifici, aquest creixement d'alumnes també afecta a altres processos interns de l'escola que tot i no ser visibles a simple vista, no deixen de ser molt importants. Un d'aquests processos és la matriculació d'alumnes.

La universitat necessita saber quina quantitat de diners ingressarà el semestre següent en concepte de matrícules i quants alumnes es matricularan a cada assignatura. Això requereix d'un software especialitzat del qual n'esteu llegint la seva memòria. Aquest aplicatiu a través de l'entrada de dades externes en format excel, és capaç de realitzar aquesta previsió que fins ara es fa manualment.

Resumen

Este año nuestra universidad EUPMT ha cambiado su ubicación por otra de mas amplia y adecuada bajo el nombre de Tecnocampus. Debido básicamente, a que la escuela ha ido creciendo y el anterior edificio ha quedado anticuado y pequeño. De la misma forma que el crecimiento de la escuela desembocó en el traslado al actual edificio, este crecimiento de alumnos también afecta a otros procesos internos de la escuela que todo y no ser visibles a simple vista, no dejan de ser muy importantes. Uno de estos procesos es la matriculación de alumnos.

La universidad necesita saber qué cantidad de dinero ingresará el semestre siguiente en concepto de matrículas i cuantos alumnos se matricularan en cada asignatura. Esto requiere de un software especializado del que estáis leyendo su memoria. Este aplicativo a través de la entrada de datos externos en formato excel, es capaz de realizar esta previsión que hasta ahora se realizaba manualmente.

Abstract

This year our University EUPMT has changed its location for another, more broader and appropriate under the name Tecnocampus. Basically due to the fact that the school has been growing and the previous building has become old fashioned and small. The growths of the school led to the relocation of the current building, affecting other internal processes not visible to the naked eye, but are important. One of these processes is the enrollment of students.

The university needs to know which is the amount of money will enter the coming semester in concept of matriculation and how many students will be enrolled in each subject. This requires specialized software, the explanation of which is in this report. This applet, through to the input of external data in excel format is capable of performing this forecast that until now was done manually.

Index

Glossari de termes.

BBDD	Bases de Dades
CD	Compact Disc
EUPMT	Escola Universitària Politècnica de Mataró
JVM	Java Virtual Machine
MVC	Model Vista Controlador
OO	Orientat a Objectes
PC	Personal Computer (Ordinador Personal)
PFC	Projecte Final de Carrera
SGBBDD	Sistema Gestor de Bases de Dades
TFC	Treball Final de Carrera

1. Introducció

1.1. Elecció del projecte

Davant d'una absoluta falta de motivació davant d'una carrera que s'havia d'haver quedat en una simple afició, vaig elegir una projecte que complís tres requisits:

- Fàcil: buscar un projecte que dintre l'elevada dedicació reclamen, en tingui la mínima.
- Sense noves tecnologies: ampliació del anterior punt, només vull aplicar el que he aprés al llarg de la carrera.
- Útil: després de dedicar unes 300 hores al actual projecte, al qual li tinc una estrany sentiment d'amor i odi, no m'agradaria que es passés la resta de la seva vida confinat en un calaix. Almenys que algú li tregui profit durant una temporada.

1.2. Motivacions

Obtenir el títol d'enginyer tècnic en informàtica de gestió per la seva posterior col·locació en el currículum.

2. Objectius.

Aquest és el projecte d'un software de gestió connectat a un SGBBDD, en aquest cas MySQL. Aquest aplicatiu permet portar el control docent de les diferents assignatures de la universitat i la creació de previsions de pressupostos i matrícules. També integra i cohesiona en un sol lloc, tota la informació utilitzada que fins ara estava dispersada en diferents llocs i formats.

A continuació els diferents objectius més detallats:

- **Gestió de les assignatures:** Totes les carreres amb les seves respectives assignatures s'han de poder gestionar des d'una mateixa pantalla.
- **Gestió del professorat:** El software permet la gestió del professorat amb les seves dades personals.
- **Gestió d'històrics:** Cada assignatura que s'ha impartit, és emmagatzemada amb el nombre d'estudiants matriculats i aprovats per estudis estadístics posteriors a l'hora de fer les previsions.
- **Gestió d'oferta:** Una oferta és una assignatura, la qual s'impartirà en un semestre i amb un professor en concret. Aquest aplicatiu permet gestionar-ho de tal manera que facilita l'assignació dels diferents professors de l'escola a diferents ofertes.
- **Gestió de matricules:** L'aplicatiu permet visionar de formar clara amb diferents formes de filtratge, les matriculacions que s'han fet en les diferents ofertes d'assignatures i els diners que s'han ingressat per aquests conceptes.
- **Previsió de matricules:** tot els objectius que s'assoleixen fins ara, tot i ser útils perquè ajuden a compondre un marc comú on centralitzar totes les dades existents, tenen com a finalitat permetre calcular al programa, les previsions de matricules de cada alumne i els ingressos que s'obtindran d'elles.

3. Anàlisi

3.1 Estudi previ del projecte

3.1.1 Situació actual de la gestió pressupostària en L'EUPMT

Actualment a la universitat no hi ha cap software específic que realitzi aquesta funció, s'utilitzen simples fulles excel. Aquestes fulles excels (les quals aquest software utilitza per extreure'n informació pel seu funcionament) disposa de taules dinàmiques per amenitzar la tasca al cap d'estudis (encarregat de dur aquesta tasca a terme). Tot i això, el cap d'estudis, ha de dedicar a anar alumne per alumne i fer una previsió de les assignatures en què es matricularà. Una tasca molt llarga i feixuga tenint en compte la quantitat d'alumnes que hi ha a l'EUPMT.

3.1.2 Conclusions

El sistema utilitzat és totalment ineficient i impropi d'una universitat que imparteix carreres tecnològiques. Es podria aplicar perfectament la dita castellana de "a casa del herrero cuchara de palo".

Una persona altament qualificada amb un sou corresponent a les responsabilitats de cap d'estudi, no pot dedicar-se a fer una tasca monòtona i repetitiva que podria fer una màquina. La universitat està malgastant recursos humans i no precisament barats, això ha de canviar.

3.2 Requisits

Els requisits (que no requeriments, degut a una mala traducció de l'anglès "requirement"), són totes les necessitats que ha de satisfer un software per poder complir els objectius marcats anteriorment.

A continuació es separen els requisits en dos: funcionals i no funcionals.

3.2.1 Funcionals

Els requisits funcionals són els defineixen el funcionament intern de l'aplicatiu, és a dir, mostren com els casos d'ús seran duts a terme.

3.2.2 No funcionals

Els requisits no funcionals són criteris imposats pel client o pel software, és a dir, es refereixen a tots els requisits que no descriuen ni funcions a realitzar ni informació a guardar.

3.3 Casos d'ús

Gestió d'assignatures

Cas d'ús: Alta assignatura

Actors: Administrador

Descripció: Cas d'ús en què es dóna d'alta una assignatura a la BBDD.

Precondicions:

Hi ha d'haver alguna carrera carregada a la BBDD.

Flux normal:

1. S'introdueixen les dades demanades pel software per poder donar d'alta a l'assignatura.
2. El sistema valida les dades.

Flux alternatiu:

1.1 S'introdueix la ruta del fitxer excel que conté les dades.

2.1 Si les dades no són correctes, es torna al pas 1 del flux normal.

Post-condicions:

Si tot és correcte, la nova assignatura està inserida a la BBDD i es tanca la finestra per retornar a l'anterior pantalla.

Cas d'ús: Baixa assignatura

Actors: Administrador

Descripció: Cas d'ús en què es dóna de baixa una assignatura de la BBDD.

Precondicions:

Hi ha d'haver alguna assignatura carregada a la BBDD.

Flux normal:

1. Es selecciona l'assignatura que es vol eliminar.
2. Es confirma la seva eliminació.

Post-condicions:

S'elimina l'assignatura de la BBDD i els seus registres relacionats.

Gestió de carreres

Cas d'ús: Alta carrera

Actors: Administrador

Descripció: Cas d'ús en què es dóna d'alta una carrera a la BBDD.

Flux normal:

1. S'introdueixen les dades demanades pel software per poder donar d'alta a la carrera.
2. El sistema valida les dades.

Post-condicions:

Si tot és correcte, la nova carrera està inserida a la BBDD i es tanca la finestra per retornar a l'anterior pantalla.

Cas d'ús: Baixa carrera

Actors: Administrador

Descripció: Cas d'ús en què es dóna de baixa una carrera de la BBDD.

Pre-condicions:

Hi ha d'haver alguna carrera carregada a la BBDD.

Flux normal:

1. Es selecciona l'assignatura que es vol eliminar.
2. Es confirma la seva eliminació.

Post-condicions:

S'elimina la carrera de la BBDD i els seus registres relacionats.

Gestió d'històrics

Cas d'ús: Alta històric

Actors: Administrador

Descripció: Cas d'ús en què es dóna d'alta un històric a la BBDD.

Precondicions:

A la BBDD han d'estar introduïts: els semestres, les assignatures amb les seves respectives carreres i els professors.

Flux normal:

1. S'introdueixen les dades demanades pel software per poder donar d'alta a l'històric.
2. El sistema valida les dades.

Flux alternatiu:

1.1 S'introdueix la ruta del fitxer excel que conté les dades.

2.1 Si les dades no són correctes, es torna al pas 1 del flux normal.

Post-condicions:

Si tot és correcte, el nou històric està inserit a la BBDD i es tanca la finestra per retornar a l'anterior pantalla.

Cas d'ús: Baixa d'històric

Actors: Administrador

Descripció: Cas d'ús en què es dóna de baixa un històric de la BBDD.

Pre-condicions:

Hi ha d'haver algun històric carregat a la BBDD.

Flux Normal:

1. Es selecciona l'històric que es vol eliminar.
2. Es confirma la seva eliminació.

Post-condicions:

S'elimina l'històric de la BBDD i els seus registres relacionats.

Gestió d'ofertes

Cas d'ús: Alta oferta

Actors: Administrador

Descripció: Cas d'ús en què es dóna d'alta una oferta a la BBDD.

Pre-condicions:

A la BBDD han d'estar introduïts: els semestres, les assignatures amb les seves respectives carreres i els professors.

Flux normal:

1. S'introdueixen les dades demanades pel software per poder donar d'alta a l'oferta.
2. El sistema valida les dades.

Flux alternatiu:

1.1 S'introdueix la ruta del fitxer excel que conté les dades.

2.1 Si les dades no són correctes, es torna al pas 1 del flux normal.

Post-condicions:

Si tot és correcte, la nova oferta està inserida a la BBDD i es tanca la finestra per retornar a l'anterior pantalla.

Cas d'ús: Baixa oferta

Actors: Administrador

Descripció: Cas d'ús en què es dóna de baixa una oferta de la BBDD.

Precondicions:

Hi ha d'haver alguna oferta carregada a la BBDD.

Flux normal:

1. Es selecciona l'històric que es vol eliminar.
2. Es confirma la seva eliminació.

Post-condicions:

S'elimina l'oferta de la BBDD i els seus registres relacionats.

Gestió de professors

Cas d'ús: Alta professor

Actors: Administrador

Descripció: Cas d'ús en què es dóna d'alta un professor a la BBDD.

Flux normal:

1. S'introdueixen les dades demanades pel software per poder donar d'alta al professor.
2. El sistema valida les dades.

Post-condicions:

Si tot és correcte, el nou professor està inserit a la BBDD i es tanca la finestra per retornar a l'anterior pantalla.

Cas d'ús: Baixa professor

Actors: Administrador

Descripció: Cas d'ús en què es dóna de baixa un professor de la BBDD.

Pre-condicions:

Hi ha d'haver algun professor carregat a la BBDD.

Flux normal:

1. Es selecciona l'assignatura que es vol eliminar.
2. Es confirma la seva eliminació.

Post-condicions:

S'elimina el professor de la BBDD i els seus registres relacionats.

Gestió de semestres

Cas d'ús: Alta semestre

Actors: Administrador

Descripció: Cas d'ús en què es dóna d'alta un semestre a la BBDD.

Flux normal:

1. S'introdueixen les dades demanades pel software per poder donar d'alta al semestre.
2. El sistema valida les dades.

Post-condicions:

Si tot és correcte, el nou semestre estarà inserit a la BBDD i es tanca la finestra per retornar a l'anterior pantalla.

Cas d'ús: Modificació semestre

Actors: Administrador

Descripció: Cas d'ús en què es modifica un semestre de la BBDD.

Pre-condicions:

Hi ha d'haver algun semestre carregat a la BBDD.

Flux normal:

1. Es selecciona el semestre a modificar.
2. Es fan les pertinents modificacions de les dades i s'accepten els canvis.
3. El sistema valida els canvis realitzats.

Post-condicions:

Si tot és correcte, els canvis realitzats es veuran reflectits a la BBDD.

Cas d'ús: Baixa semestre

Actors: Administrador

Descripció: Cas d'ús en què es dóna de baixa un semestre de la BBDD.

Pre-condicions:

Hi ha d'haver algun semestre carregat a la BBDD.

Flux normal:

1. Es selecciona el semestre que es vol eliminar.
2. Es confirma la seva eliminació.

Post-condicions:

S'elimina el semestre de la BBDD i els seus registres relacionats.

Gestió d'altres

Cas d'ús: Modificació d'altres

Actors: Administrador

Descripció: Cas d'ús en què es modifiquen paràmetres necessaris pel funcionament del software.

Flux normal:

1. Es fan les pertinents modificacions de les dades i s'accepten els canvis.
2. El sistema valida els canvis realitzats.

Post-condicions:

Si tot és correcte, els canvis realitzats es veuran reflectits a la BBDD.

Gestió de matrícules

Cas d'ús: Carregar dades

Actors: Administrador

Descripció: Cas d'ús en què es carreguen a la BBDD les dades de les matriculacions actuals.

Pre-condicions:

A la BBDD han d'estar carregades totes les dades, ja que fins ara, tots les dades inserides serveixen per mostrar les matrícules i fer-ne una previsió.

Flux normal:

1. S'indica la ruta del fitxer excel amb les dades de les matrícules.
2. El sistema valida les dades.

Flux alternatiu:

2.1 Si les dades no són correctes, es torna al pas 1 del flux normal.

Post-condicions:

Si tot és correcte, els canvis realitzats es guarden a la BBDD i es mostren per pantalla.

Cas d'ús: Realitzar previsions de matrícules

Actors: Administrador

Descripció: S'examinaran tots els alumnes i es farà una previsió de quines assignatures es matricularan. Això es mostrarà per pantalla agrupant els alumnes en les diferents ofertes d'assignatures i es calcularà els ingressos que generaran a la universitat.

Pre-condicions:

A la BBDD han d'estar carregades les dades referents a l'actual matrícula.

Flux normal:

1. L'administrador selecciona l'opció de realitzar la previsió.
2. L'administrador confirma l'ordre, ja que aquest procés pot durar uns minuts.
3. El sistema realitza les previsions.

Post-condicions:

La previsió es mostra per pantalla i els resultats es guarden a la BBDD perquè no s'hagi de tornar a realitzar el càlcul.

3.4 Anàlisi de les eines i tecnologies usades

3.4.1 Tecnologies

JAVA: és un llenguatge de programació orientat a objectes i interpretat. OO perquè posseeix les característiques d'aquests com són els objectes, classes i subclasses, herència, enllaços dinàmics i encapsulament. Interpretat perquè el codi compilat no pot ser executat directament per una màquina, aquesta ha de disposar del JVM (Java Virtual Machine), que interpreta el codi del programa i l'executa en la màquina en qüestió. Això tot i semblar un handicap, és un dels trets característics d'aquest llenguatge ja que permet que un mateix codi sigui executat en qualsevol plataforma sense afectar la seva velocitat d'execució.

SQL: és un llenguatge declaratiu d'accés a BBDD relacionals. És a dir, un llenguatge estandarditzat que permet comunicar-se amb la majoria de BBDD per tal d'extreure'n la informació que contenen. Aquest llenguatge és el que utilitza el SGBBDD emprat en el TFC, el MySQL.

3.4.2 Eines

NetBeans 6.8: És una aplicació open source pel desenvolupament d'aplicacions en diferents llenguatges (JAVA, Ruby, UML, C++...), però sobretot el JAVA. Conté una gran varietat de mòduls per l'ampliació d'aquest entorn tot i que en aquest cas, la versió per defecte ha estat suficient. Ha estat l'eina més utilitzada per dur a terme el PFC, des de les seves etapes inicials pel disseny de diagrames fins a l'última per realitzar els tests.

MySQL: és un SGBBDD molt complet que destaca per la seva estabilitat, escalabilitat i està disponible a diferents plataformes. Que es distribueixi sota la llicència GPL (excepte per a productes privatius, llavors s'ha d'abonar una llicència) l'han convertit en un dels SGBBDD més populars.

Tots i que en un principi aquest projecte es va idear per funcionar sota Microsoft Acces, ja que per una BBDD senzilla i monousuari en els requisits previs feia creure que seria suficient. Posteriors tests van demostrar la faltar de potència de l'Access i la necessitat de trobar un

SGBBDD més potent que satisfés les necessitats d'aquest software i l'escollit va ser MySQL per les raons anteriorment esmenades.

Adobe Photoshop CS 4: Software molt potent per l'edició gràfica i el millor per la gran majoria de professionals. Tot i que el seu ús en el PFC ha estat testimonial, s'ha inclòs per la seva gran utilitat. A l'hora de modificar algunes imatges per aconseguir el logotip del tecnocampus i les diferents icones que es poden observar.

4 Disseny

4.1 Disseny de l'aplicació

L'aplicació ha estat dissenyada seguint el patró MVC (Model Vista-Controlador). Aquest patró separa en tres components diferents: les dades de l'aplicació, la interfície gràfica que veu l'usuari i la lògica de control.

Per realitzar-ho, s'han implementat un total de cinc capes:

- Aplicació: on anirà el controlador del software.
- Domini: per representar les entitats de la BBDD i poder treballar en elles.
- Persistència: on aniran les classes que interaccionaran amb la BBDD.
- Presentació: on aniran totes les pantalles i la part gràfica del software.
- Utils: un calaix de sastre on aniran classes útils (valgui la redundància) per diferents problemes que aniran sorgint al llarg del desenvolupament del programa.

A continuació, a cada apartat s'escriuran les diferents classes amb els procediments i funcions més representatius. Tot i semblar excessiu, s'ha considerat adient ja que aquest projecte serà llegat a la universitat (codi font inclòs) per la seva futura modificació si així es desitja. Per tan, una explicació extensa de totes les classes facilitarà la comprensió del software a qui hagi de treballar en ell.

4.2 Disseny de la capa presentació

En aquesta capa, hi ha una classe per cada pantalla que es mostra, són les següents:

- Inici
- Pantalla_Altres
- Pantalla_Assignatures
- Pantalla_Assignatures_add
- Pantalla_Carrera
- Pantalla_Carrera_add

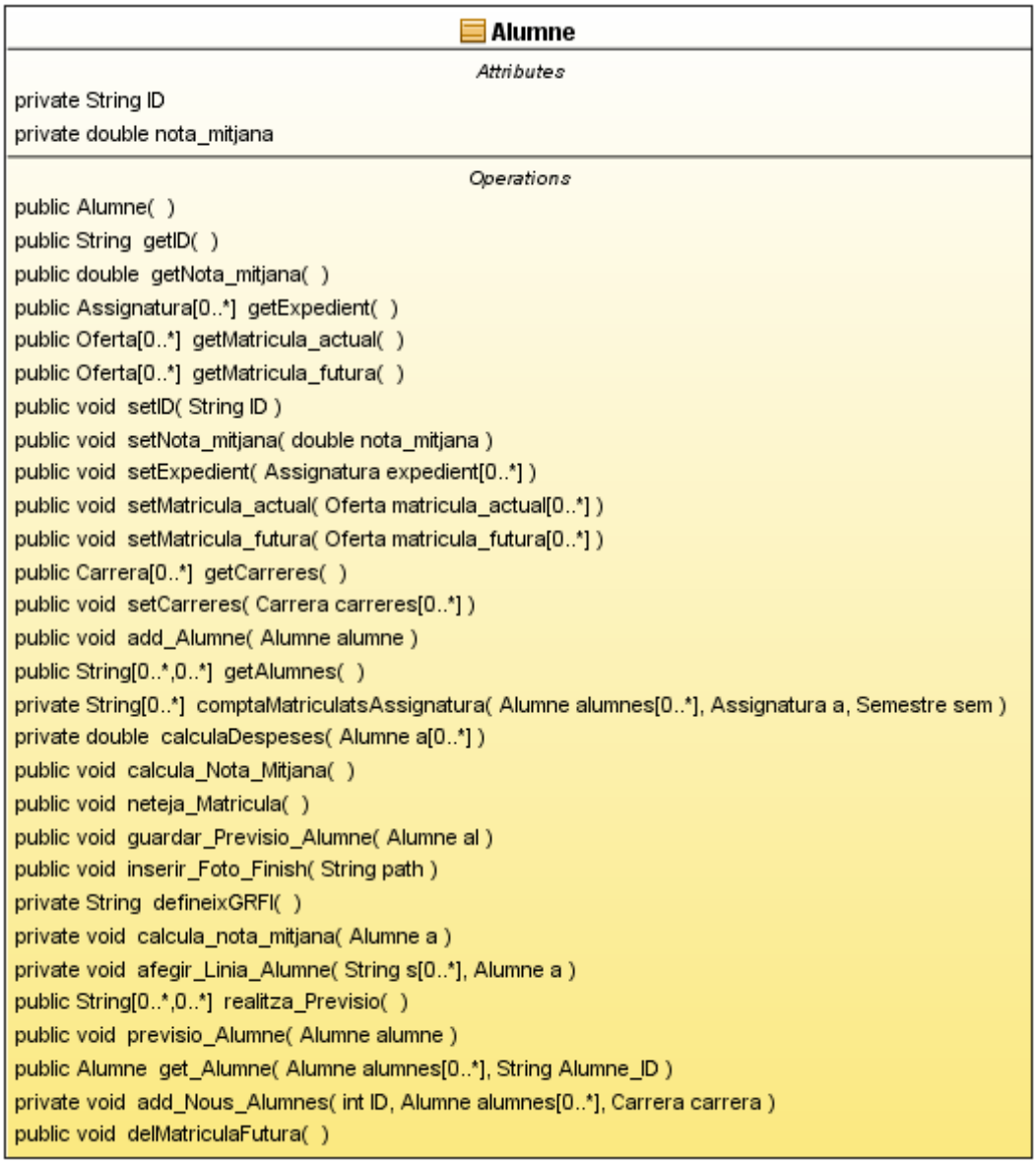
- Pantalla_Historic
- Pantalla_Historic_add
- Pantalla_Oferta
- Pantalla_Oferta_add
- Pantalla_Pressupostador
- Pantalla_Profe
- Pantalla_Profe_add
- Pantalla_Semestre
- Pantalla_Semestre_add

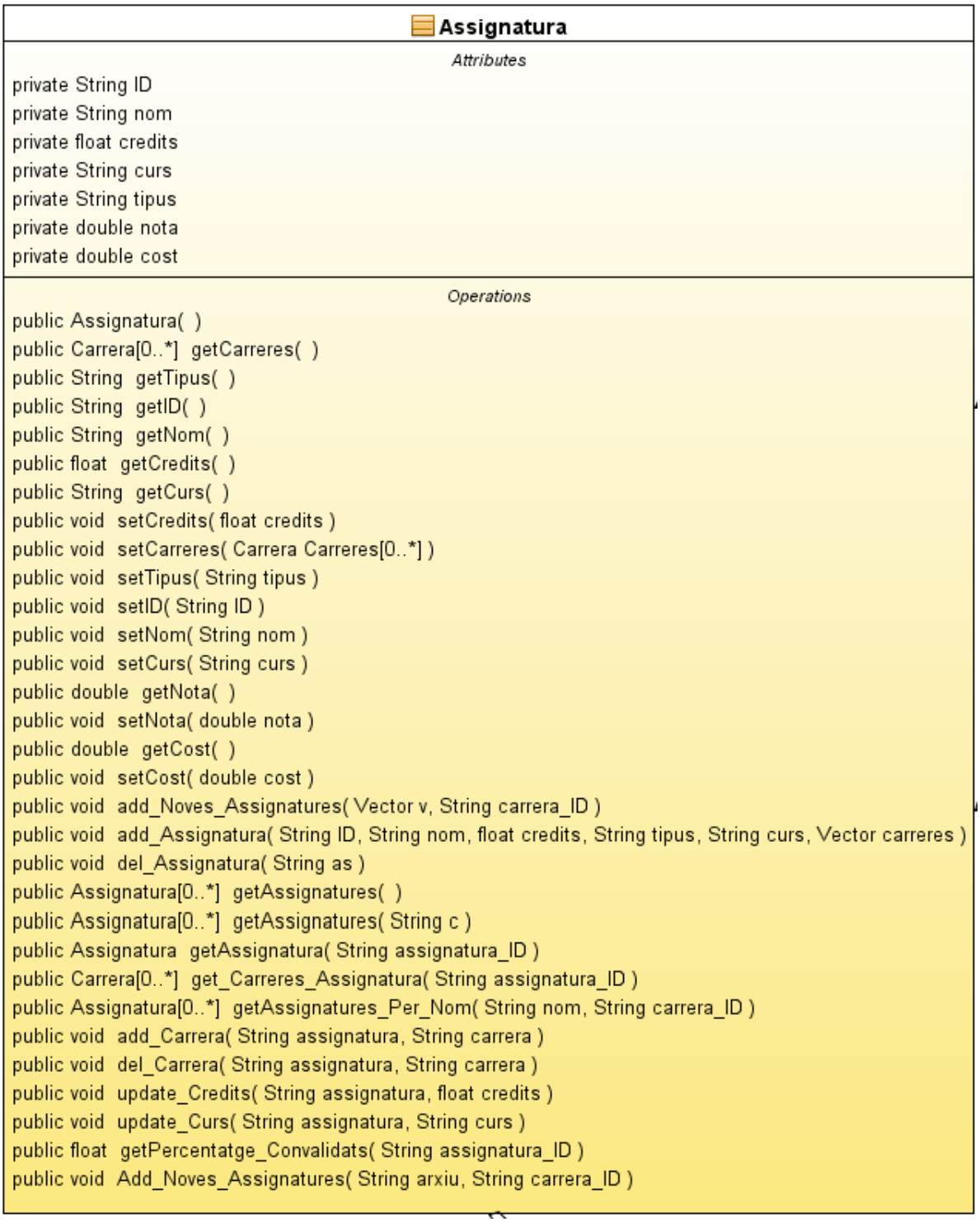
Totes aquestes classes són molt semblants en quan a mètodes. Quasi totes tenen el procediment carregaTaula(), que carrega a la graella les dades demanades per l'usuari.

4.3 Disseny de la capa aplicació

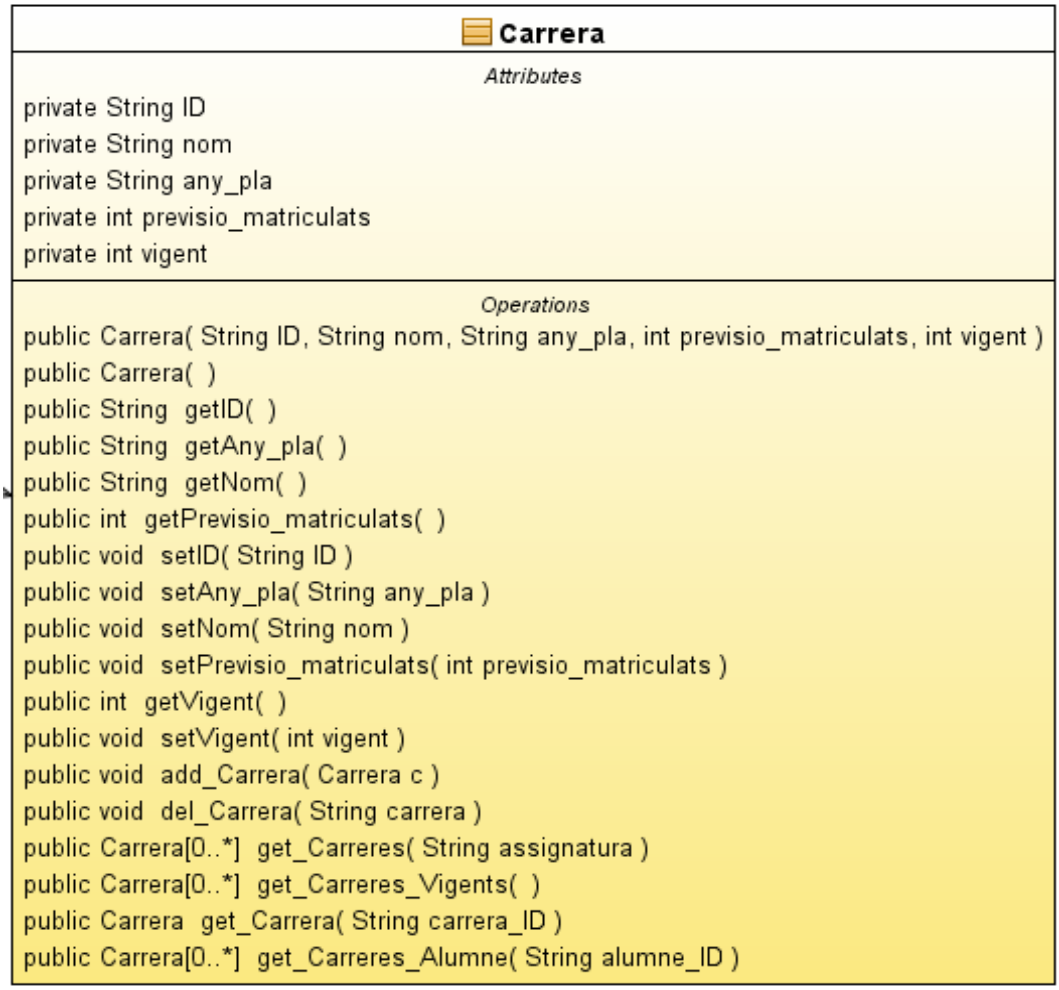
Controlador: Aquesta classe és un mer enllaç entra la capa presentació i domini. Les seves funcions són transmetre la informació que envia l'usuari cap a capes inferiors i convertir la informació que rep de l'aplicació cap a presentació transformant les dades en classes més simples com strings.

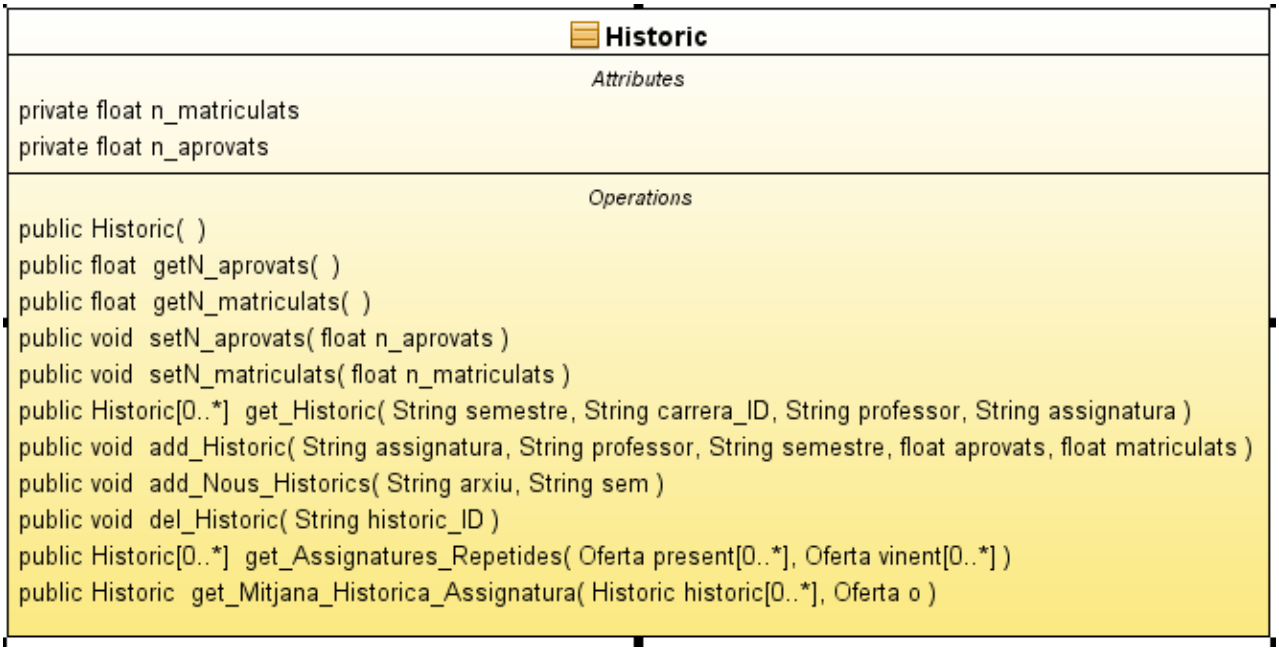
4.4 Disseny de la capa domini





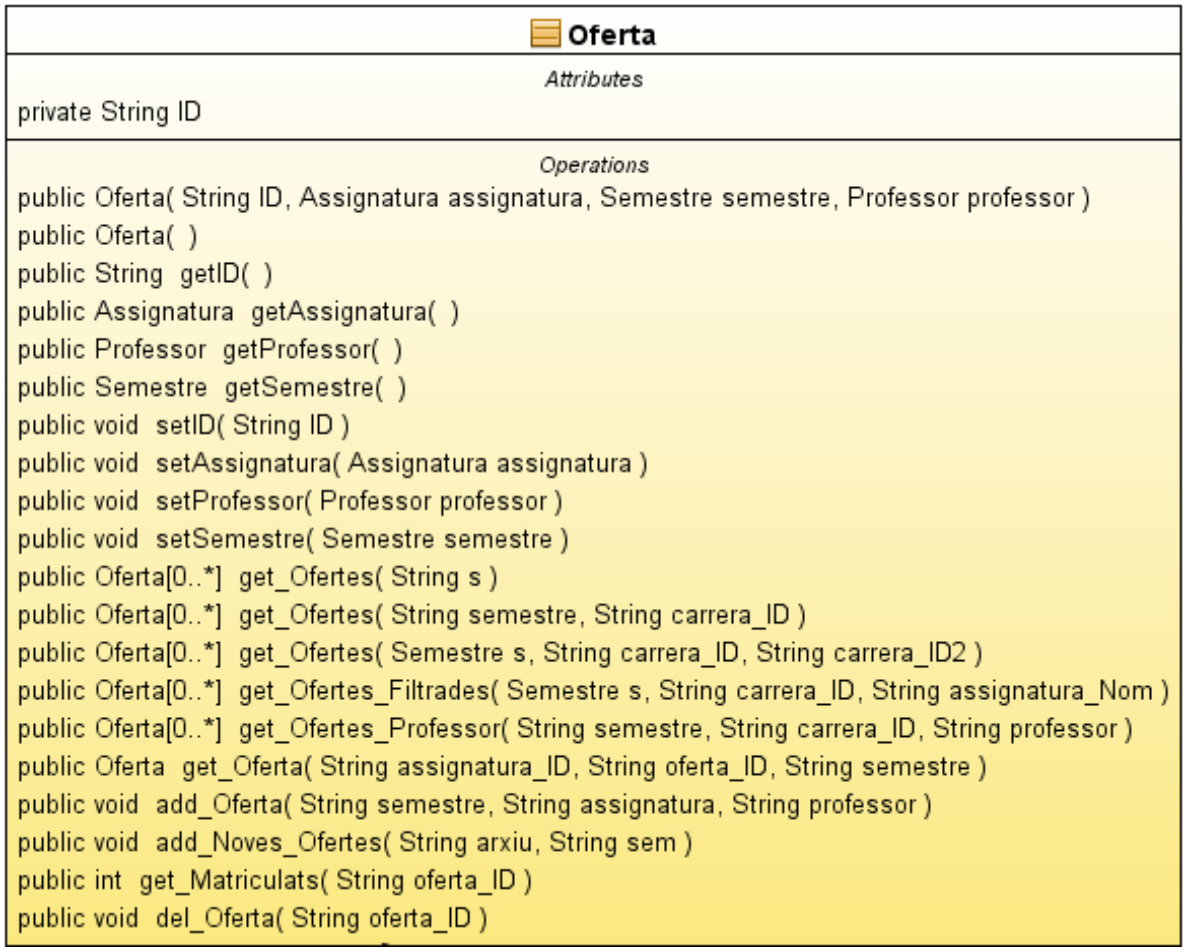
- **Add_Noves_Assignatures**(String arxiu): procediment, al qual se li passa la ruta d'un fitxer en format "csv" o "xls" amb el nou pla d'estudis d'alguna carrera. Segons el tipus d'arxius, utilitza les classes CVS_Loader o XLS_Loader per carregar les noves assignatures amb un vector i passar-ho a persistència.

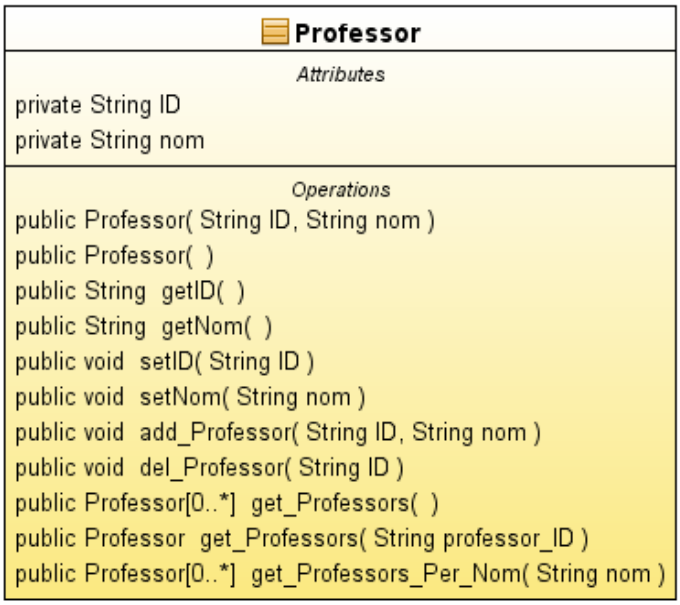

Carrera
Attributes
private String ID
private String nom
private String any_pla
private int previsio_matriculats
private int vigent
Operations
public Carrera(String ID, String nom, String any_pla, int previsio_matriculats, int vigent)
public Carrera( )
public String getID( )
public String getAny_pla( )
public String getNom( )
public int getPrevisio_matriculats( )
public void setID(String ID)
public void setAny_pla(String any_pla)
public void setNom(String nom)
public void setPrevisio_matriculats(int previsio_matriculats)
public int getVigent( )
public void setVigent(int vigent)
public void add_Carrera(Carrera c)
public void del_Carrera(String carrera)
public Carrera[0..*] get_Carreres(String assignatura)
public Carrera[0..*] get_Carreres_Vigents( )
public Carrera get_Carrera(String carrera_ID)
public Carrera[0..*] get_Carreres_Alumne(String alumne_ID)

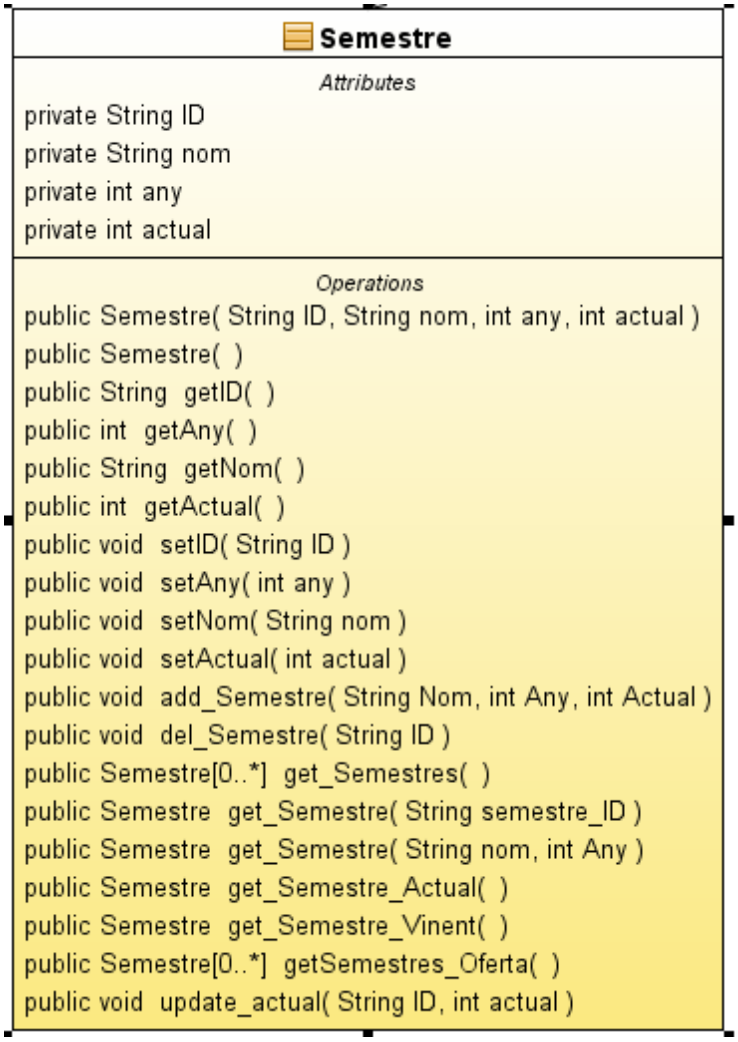

Historic
Attributes
private float n_matriculats
private float n_aprovats
Operations
public Historic( )
public float getN_aprovats( )
public float getN_matriculats( )
public void setN_aprovats(float n_aprovats)
public void setN_matriculats(float n_matriculats)
public Historic[0..*] get_Historic(String semestre, String carrera_ID, String professor, String assignatura)
public void add_Historic(String assignatura, String professor, String semestre, float aprovats, float matriculats)
public void add_Nous_Historics(String arxiu, String sem)
public void del_Historic(String historic_ID)
public Historic[0..*] get_Assignatures_Repetides(Oferta present[0..*], Oferta vinent[0..*])
public Historic get_Mitjana_Historica_Assignatura(Historic historic[0..*], Oferta o)

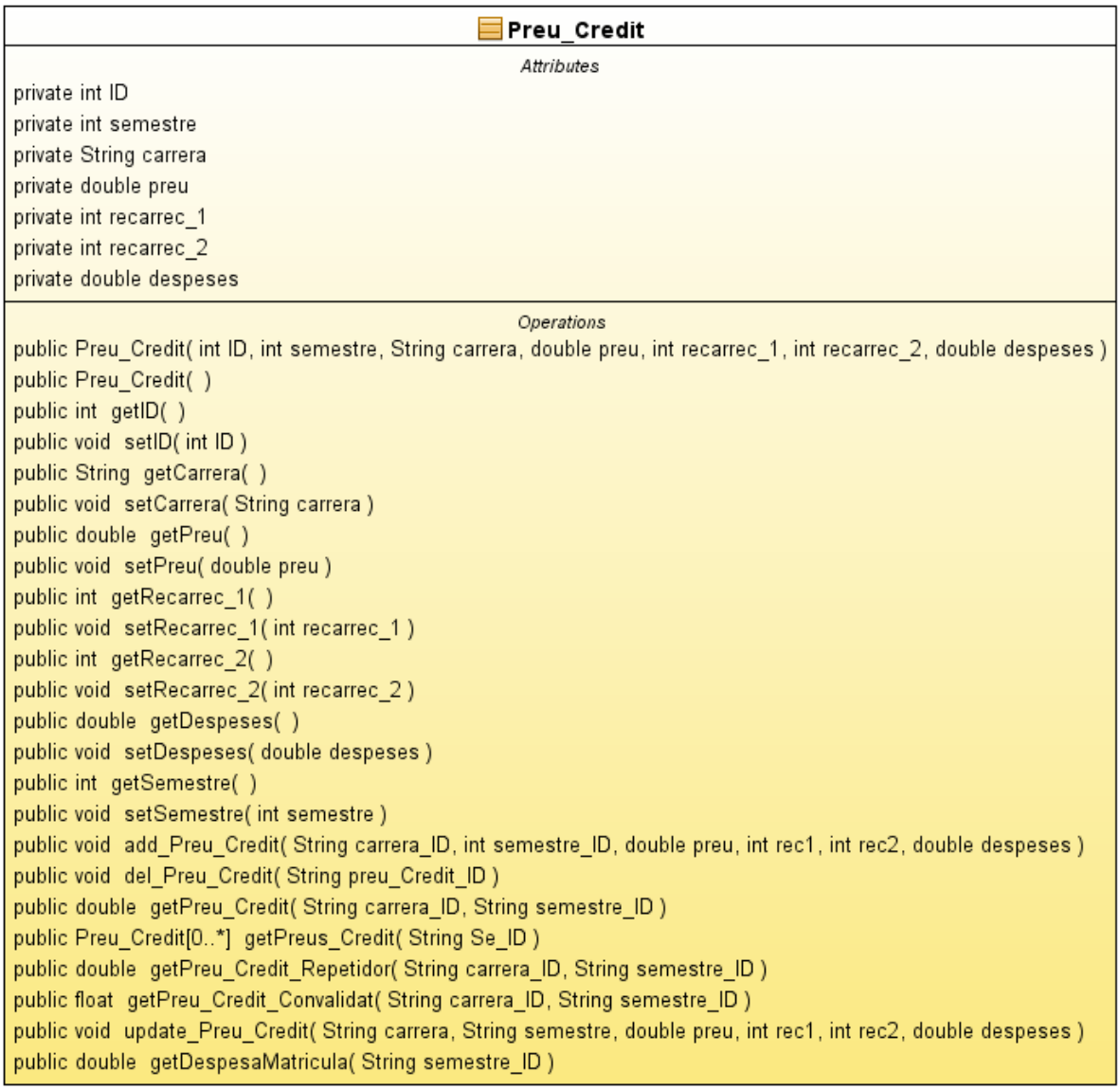
- **get_Assignatures_Repetides(**ArrayList<Oferta> present, ArrayList<Oferta> vinent**):** funció que retorna un ArrayList d'històric d'assignatures, amb les assignatures que es repeteixen de l'actual i següent semestre.

- **get_Mitjana_Historica_Assignatura(**ArrayList<Historic> historic, Professor p**):** funció que retorna un historic d'assignatura amb les variables d'aprovats i matriculats com a mitjana de totes les vegades que ha fet l'assignatura durant els últims deu anys. Si és la primera vegada que el professor fa aquesta assignatura, la mitjana es realitza independentment del professor, si és la primera vegada que es fa l'assignatura, es posa una aproximació indicada pel gestor del software.




Professor
Attributes
private String ID
private String nom
Operations
public Professor(String ID, String nom)
public Professor( )
public String getID( )
public String getNom( )
public void setID(String ID)
public void setNom(String nom)
public void add_Professor(String ID, String nom)
public void del_Professor(String ID)
public Professor[0..*] get_Professors( )
public Professor get_Professors(String professor_ID)
public Professor[0..*] get_Professors_Per_Nom(String nom)


Semestre
Attributes
private String ID
private String nom
private int any
private int actual
Operations
public Semestre(String ID, String nom, int any, int actual)
public Semestre( )
public String getID( )
public int getAny( )
public String getNom( )
public int getActual( )
public void setID(String ID)
public void setAny(int any)
public void setNom(String nom)
public void setActual(int actual)
public void add_Semestre(String Nom, int Any, int Actual)
public void del_Semestre(String ID)
public Semestre[0..*] get_Semestres( )
public Semestre get_Semestre(String semestre_ID)
public Semestre get_Semestre(String nom, int Any)
public Semestre get_Semestre_Actual( )
public Semestre get_Semestre_Vinent( )
public Semestre[0..*] getSemestres_Oferta( )
public void update_actual(String ID, int actual)

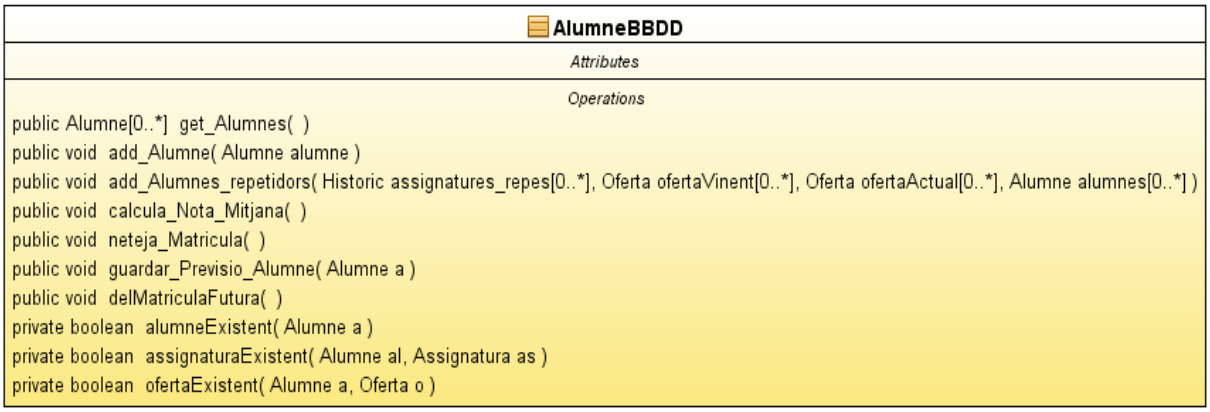


Tot i no ser pròpiament del domini, insereixo aquestes dues classes de la capa Utils aquí, ja que és amb aquesta capa amb qui únicament interaccionen.

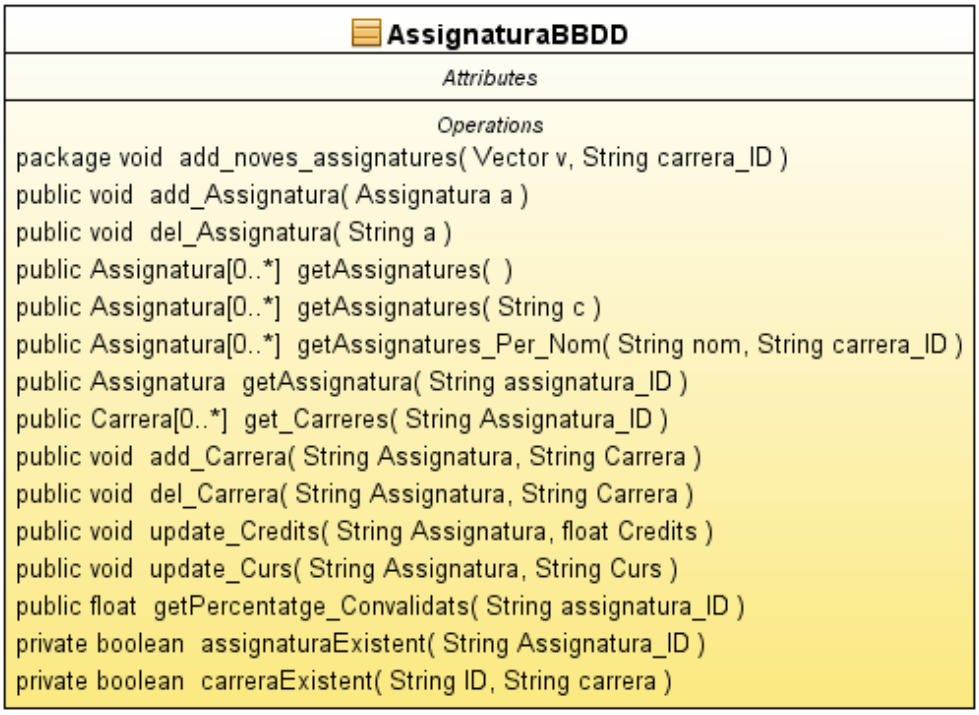
- **CVS_Loader:**
  - **CVS_Convert**(String arxiu): Els fitxers "CSV" ja són fitxers que van delimitats per punts i comes, per tan l'únic que fa aquesta funció és llegir-lo i convertir-lo en un vector per poder treballar amb les dades posteriorment.

- **XLS_Loader:**
  - **XLS_Convert**(String arxiu): funció que retorna un vector. Utilitzant la llibreria jexcel, llegeix un fitxer excel en format "xls". Va recorrent les files de la fulla de càlcul i les emmagatzema en un string al vector. Aquest string està separat en diferents parts per punts i comes per delimitar les cel·les de la fulla excel.
  - **excels_Opener(**String ruta**):** aquest procediment serveix per obrir qualsevol fulla excel passant-li la seva ruta en un string com a paràmetre. S'utilitzarà per obrir les fulles d'excel de mostra, per saber com s'han de col·locar les dades a les fulles de càlcul a l'hora que el software hagi de fer migracions de dades.

4.5 Disseny de la capa persistència

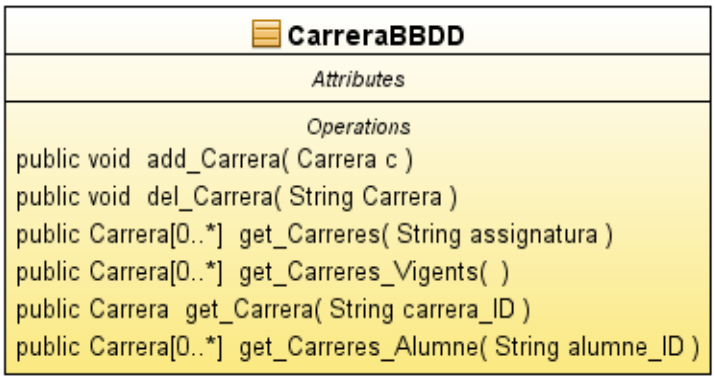


- **calcula_Nota_Mitjana():** procediment que calcula la nota mitjana dels alumnes a partir de la taula matricula i la insereix a la taula alumne de la BBDD.
- **add_Alumnes_repetidors(**ArrayList<Historic> assignatures_repes, ArrayList<Alumne> alumnes**):** procediment que se li passen dos paràmetres: un arraylist amb tots els alumnes i un arraylist amb totes les assignatures que es tornen a fer el semestre vinent respecte l'actual. Aquest procediment calcula la tassa de repetidors de l'assignatura tenint en compte el professor que l'ha exercit. Una vegada obtingut el percentatge, s'agafaran els alumnes amb pitjor nota i dins l'array de matricula futura s'afegirà aquesta oferta d'assignatura del semestre vinent.
- **get_Alumnes():** funció que carrega en un arraylist tots els alumnes i el retorna.
- **add_Alumne(**Alumne alumne, String carrera_ID**):** procediment que afegeix a la BBDD un alumne.
- **neteja_Matricula():** procediment que esborra tots els registres de la taula matricula.
- **guardar_Previsio_Alumne(**Alumne a**):** procediment que guarda a la taula matricula la previsió de matricula que escollirà l'alumne que se li passa com a paràmetre.



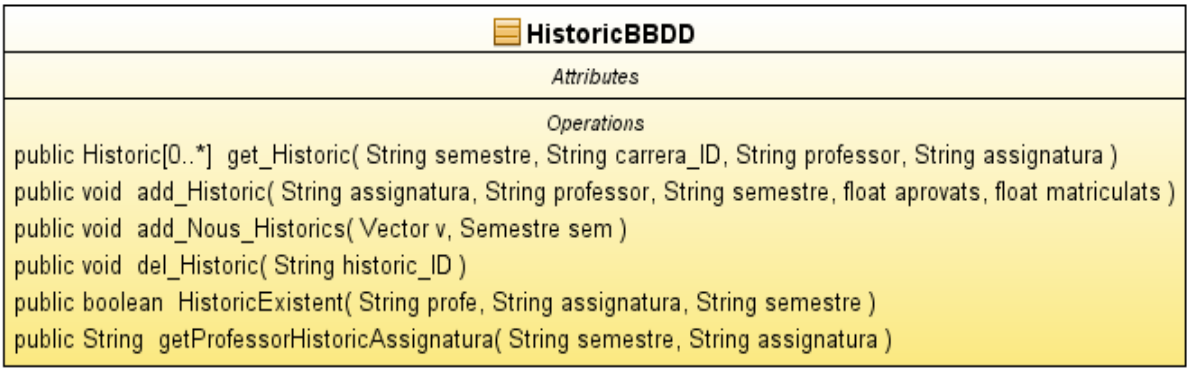
- **add_noves_assignatures(**Vector v**):** procediment que recorre el vector amb les assignatures i les va inserint a la BBDD.
- **getAssignatures()**: funció que retorna un vector amb totes les assignatures existents.
- **assignaturaExistent(**String assignatura**):** funció booleana i privada de la classe. S'utilitza en el procediment add_noves_assignatures de la següent manera, compara que l'assignatura que se li passa com a argument no existeixi. Si ja existeix, vol dir que ja s'ha registrat anteriorment perquè també es cursa en alguna altre carrera i per tan no cal tornar-la afegir. Només s'ha de guardar a la taula AsCa per tal d'indicar que l'assignatura també es cursa en la carrera a la qual estem afegint assignatures.
- **add_Assignatura(**Assignatura a**):** procediment al qual se li passa una variable de la classe assignatura perquè l'insereixi a la taula d'assignatures i també a la taula AsCa per tal d'indicar quines carreres la cursen.
- **del_Assignatura(**Assignatura a**):** procediment al qual se li passa una variable de la classe assignatura perquè esborri de les taules assignatures i AsCa totes les referències a ella.
- **getAssignatures():** funció que retorna un array list amb totes les assignatures que s'estan cursant a la universitat.

- **getAssignatures(**String carrera_ID**):** funció que retorna un array list amb totes les assignatures d'una carrera en concret que es passa com a paràmetre (l'identificador).
- **getAssignatura(**String assignatura_ID**):** retorna l'assignatura amb l'identificador que se li ha passat com a argument.
- **get_Carreres(**String Assignatura_ID**):** funció que retorna un array list amb totes les carreres que tenen una assignatura en concret que és passada com a argument.
- **add_Carrera(**String Assignatura, String Carrera**):** procediment que afegeix a una carrera, una assignatura. Només modifica la taula AsCa de la BBDD.
- **void del_Carrera(**String Assignatura, String Carrera**)**: procediment que elimina una assignatura. Només modifica la taula AsCa de la BBDD.
- **update_Credits(**String Assignatura, float Credits**)**: procediment que modifica la quantitat de crèdits que hi ha en una assignatura.
- **update_Curs(**String Assignatura, String Curs**):** procediment que modifica el semestre en què s'ha de cursar l'assignatura (1A,1B,2A,2B,3A,3B, 4A i 4B).
- **getAssignatures_Per_Nom(**String nom, String carrera_ID**):** funció que retorna un array list amb les assignatures que tinguin un nom que comenci pel string que se li passa com a paràmetre. També hi ha l'opció de filtrar-ho per carreres pel paràmetre que se li passa, si aquest és null, no filtrarà per carreres.
- **getPercentatge_Convalidats(**String assignatura_ID**):** funció que retorna el percentatge de convalidats d'una assignatura.

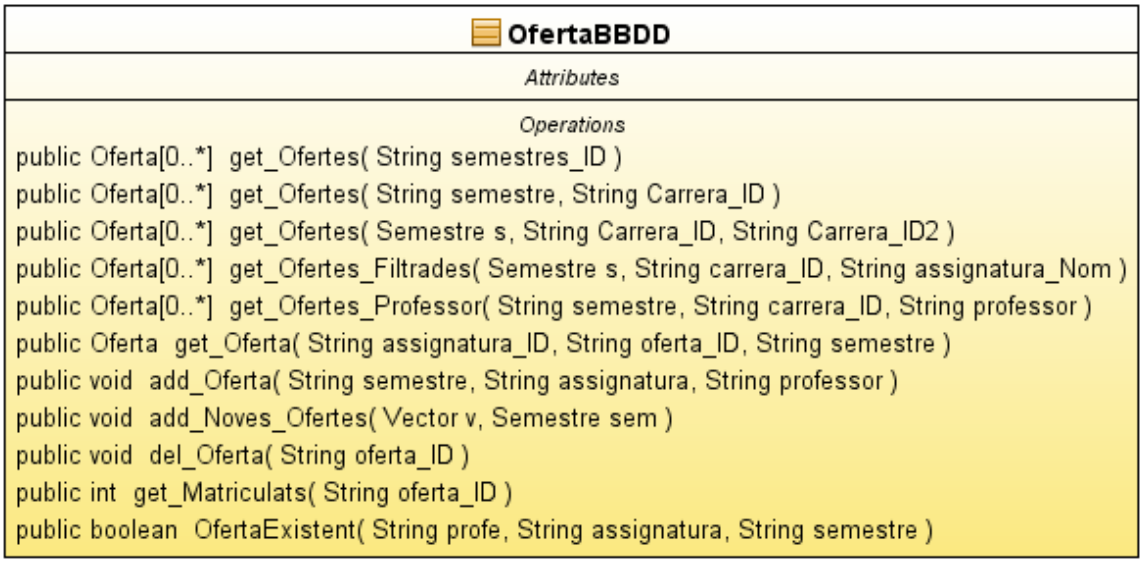


- **add_Carrera(**Carrera c**):** procediment que afegeix una carrera que li passem com a paràmetre a la taula carrera de la BBDD.

- **del_Carrera(**String Carrera**):** procediment que elimina una carrera de la BBDD. El paràmetre que se li passa és el identificador de la carrera.
- **get_Carreres(**String any**):** funció que retorna en un array list totes les carreres que es realitzen a l'universitat. El paràmetre serveix per indicar-li l'any per si volem un pla en concret, si el paràmetre està null, retornarà totes.
- **get_Carrera(**String carrera_ID**):** funció que retorna una carrera passant-li com a argument l'identificador.
- **get_Carreres_Vigents():** funció que retorna un array list amb totes les carreres dels plans d'estudi vigents.
- **get_Carreres_Alumne(**String alumne_ID): funció que retorna en un ArrayList les carreres que està realitzant un alumne. Tot i que el més usual és que només se'n realitzi una, hi ha alumnes amb doble titulació.

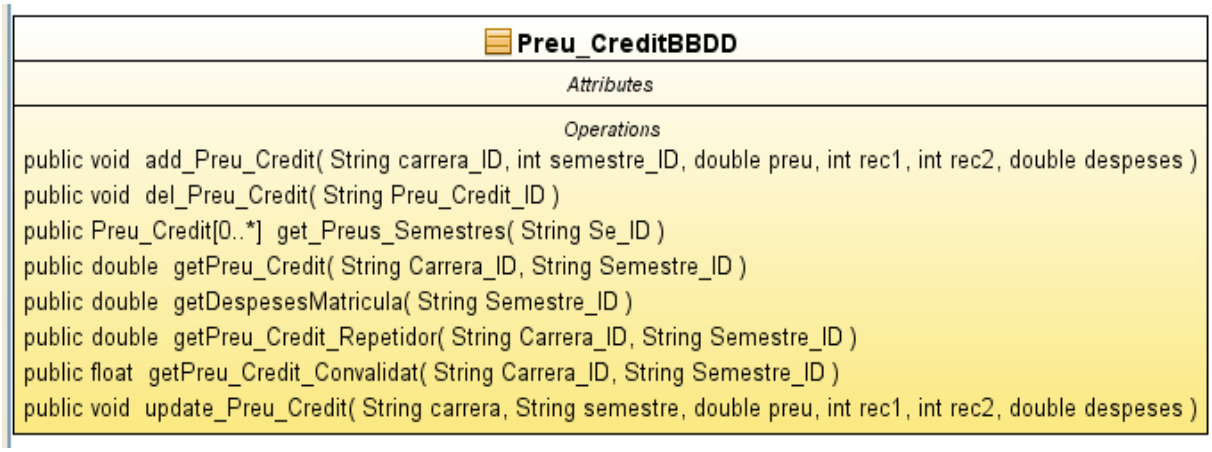


- **get_Historic(**String semestre, String carrera_ID, String professor, String assignatura**):** retorna un array list amb l'històric de totes les assignatures que s'han impartit. Se li passen quatre paràmetres per poder realitzar la cerca. Si carrera, professor o assignatura són nuls, en comptes de buscar un històric en concret, els busca tot. Si semestre és nul, buscarà tots els històrics de fa com a màxim deu anys.
- **add_Historic(**Historic h**):** afegeix un històric d'assignatura que se li passa com a paràmetre a la BBDD.
- **add_Nous_Historics(**Vector v, Semestre sem**):** afegeix a la BBDD, una fulla d'excel amb tots els històrics d'assignatures convertits en vector gràcies al mètode de la classe XLS_Convert, en el semestre indicat pel paràmetre que se li passa.
- **del_Historic(String historic_ID):** elimina de la BBDD un històric d'una assignatura amb l'identificador que se li passa com a paràmetre.

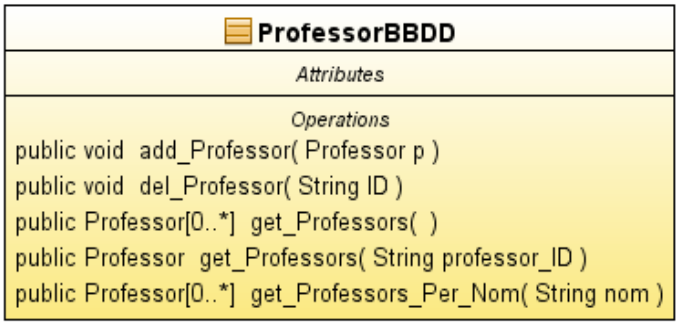


- **get_Ofertes(**Semestre s**):** retorna un array list amb totes les ofertes d'assignatures que s'imparteixen en un semestre que se li passa com a paràmetre.
- **get_Ofertes(**Semestre s, String Carrera_ID**):** retorna un array list amb totes les ofertes d'assignatures que s'imparteixen en un semestre, d'una carrera en concret. Se li passen dos paràmetres per poder realitzar la cerca.
- **get_Ofertes(**Semestre s, String Carrera_ID, String Carrera_ID2**):** retorna un array list amb totes les ofertes d'assignatures que s'imparteixen en un semestre, de dues carreres en concret, és útil pels alumnes amb doble titulació. Se li passen tres paràmetres per poder realitzar la cerca.
- **get_Ofertes_Filtrades(**Semestre s, String carrera_ID, String assignatura_Nom**):** retorna un array list amb totes les ofertes d'assignatures que s'imparteixen en un semestre, d'una carrera en concret, amb les assignatures filtrades per nom. Se li passen tres paràmetres per poder realitzar la cerca.
- **get_Oferta(**String assignatura_ID, String oferta_ID**):** retorna l'oferta d'una assignatura que s'estigui realitzant en aquest semestre. Per obtenir-la li passem com a paràmetres l'assignatura en qüestió o el propi identificador de l'oferta. Un dels dos paràmetres és null ja que per obtenir-la només en fa falta un.
- **add_Oferta(**Oferta o**):** afegeix una oferta que se li passa com a paràmetre a la BBDD.
- **add_Noves_Ofertes(**Vector v, Semestre sem**):** afegeix a la BBDD, una fulla d'excel amb totes les ofertes convertida en vector gràcies al mètode de la classe XLS_Convert, en el semestre indicat pel paràmetre que se li passa.

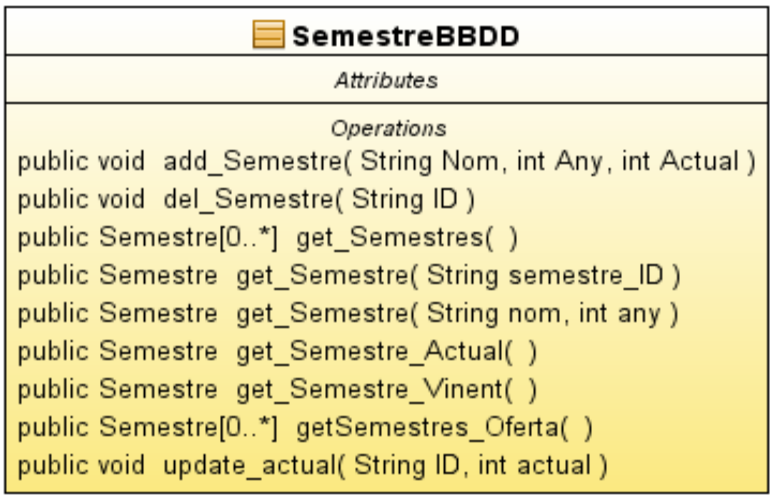
- **del_Oferta(**String oferta_ID**):** elimina de la BBDD una oferta amb l'identificador que se li passa com a paràmetre.
- **get_Matriculats(**String oferta_ID**):** retorna el nombre de matriculats d'una assignatura de l'actual semestre.



- **add_Preu_Credit(**Preu_Credit p**):** procediment que afegeix a la BBDD el preu d'un crèdit, d'una carrera en un semestre en concret.
- **del_Preu_Credit(**String Preu_Credit_ID**):** procediment que elimina de la BBDD el preu d'un crèdit d'una carrera en un semestre en concret.
- **getPreu_Credit(**String Carrera_ID, String Semestre_ID**):** funció que retorna el preu d'un crèdit d'una carrera en un semestre determinat.
- **get_Preus_Semestres(**String Se_ID**):** funció que retorna un arraylist amb els preus i recàrrecs de les diferents carreres en un semestre que se li passa com a paràmetre.
- **getPreu_Credit_Repetidor(**String Carrera_ID, String Semestre_ID**):** funció que retorna el preu d'un crèdit amb d'una carrera en un semestre determinat, amb el corresponent recàrrec per haver-la repetit.
- **getPreu_Credit_Convalidat(**String Carrera_ID, String Semestre_ID**):** funció que retorna el preu d'un crèdit d'una carrera en un semestre determinat, amb el corresponent descompte al ser convalidada per un CFGS.
- **update_Preu_Credit(**String carrera, String semestre, Float preu, int rec1, int rec2, int rec3**):** procediment que actualitza els preus i els recàrrecs d'una carrera en un semestre en concret.

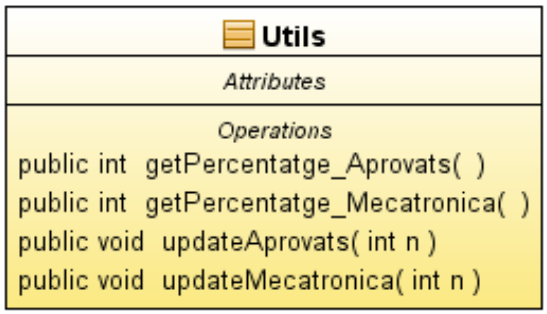


- **add_Professor(**Professor p**):** procediment que afegeix un professor que li passem com a paràmetre a la taula professor de la BBDD.
- **del_Professor(**String ID**):** procediment que elimina un professor de la BBDD. El paràmetre que se li passa és el identificador de la carrera.
- **get_Professors()**: funció que retorna en un array list tots els professors que hi ha a la universitat.
- **get_Professors(**String professor_ID**):** funció que retorna el professor indicat en el paràmetre que li passem.
- **get_Professors_Per_Nom(**String nom**):** funció que retorna un array list amb tots els professors que els seus noms comencen pel string que se li passa com a paràmetre.

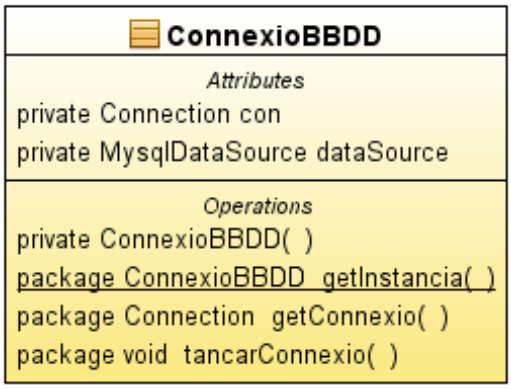


- **add_Semestre(**Semestre s**):** procediment que afegeix un semestre que li passem com a paràmetre a la taula semestre de la BBDD.

- **del_Semestre(**String ID**):** procediment que elimina un semestre de la BBDD. El paràmetre que se li passa és el semestre de la carrera.
- **get_Semestres():** funció que retorna en un array list tots els semestres que hi ha a la universitat.
- **update_actual(**String ID, int actual**):** procediment que modifica si un semestre és l'actual o no. Controla que només hi hagi un semestre com a actual.
- **get_Semestre_Actual():** funció que retorna el semestre que s'està cursant actualment.
- **get_Semestre_Vinent():** funció que retorna el semestre següent del que s'està cursant actualment.
- **getSemestres_Oferta() :** funció que retorna un arraylist amb tres semestres, l'actual i els dos vinents. És útils per utilitzar-ho en les ofertes on només es necessiten aquests tres semestres.
- **get_Semestre(**String nom, int any**):** funció que retorna un semestre passant-li com arguments l'any i si és de primavera o de tardor.

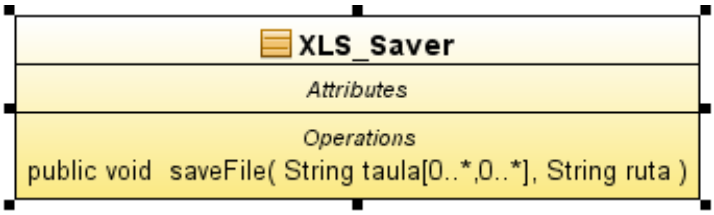


- **getPercentatge_Aprovats():** funció que retorna un enter amb el percentatge d'aprovats per assignatures que es fan per primera vegada, ho indica el gestor del software manualment.
- **getPercentatge_Mecatronica():** funció que retorna un enter amb el percentatge d'estudiants que al segon cicle es decantaran per mecatrònica en comptes d'electrònica industrial.
- **updateAprovats(**int n**):** procediment que actualitza el percentatge d'aprovats en assignatures que es fan per primera vegada.
- **updateMecatronica(**int n**):** procediment que actualitza el percentatge d'alumnes que es decantaran per mecatrònica.



- **Fassana:**
  - Aquest patró té totes les funcions i procediments de les diferents classes de persistència perquè les classes del domini només hagin d'accedir a aquesta classe i no a totes, ajudant en la claredat del codi. Per tan, no ho tornaré a escriure tot ja que seria redundant.

Tot i no està inclosa a persistència, degut a que la seva funció és manipular dades de la BBDD, es col·loca la seva explicació aquí.



  - **saveFile(**String taula[][], String ruta**):** procediment que guarda en un fitxer excel (xls) la taula que se li passa com a paràmetre, a la ruta que també se li indica en el string.

4.6 Disseny de la BBDD

Hi ha un total de 12 entitats:

- **Alumne:** taula on es registren tots els alumnes de l'actual semestre de la universitat. Té tres atributs:
  - Al_ID: identificador, per motius de la llei de protecció de dades, no es pot utilitzar ni el nom ni cognoms, per tan s'utilitzarà el seu DNI.
  - Nota mitjana: nota mitjana de l'alumne en les assignatures ja superades. Per no desvirtuar-la, no es tindran en compte les convalidades per venir de mòduls, ja que aquestes tenen per defecte una nota de 5 i desvirtua la nota, al no reflectir l'autèntic rendiment de l'alumne.

- **Professor:** taula on es guarden els diferents professor de la universitat. Té dos atributs:
  - Pr_ID: un identificador.
  - Pr_Nom: el nom del professor

- **Assignatura:** taula on es posen les diferents assignatures que imparteix el centre, té tres atributs:
  - As_ID: codi de l'assignatura, ve donat per la universitat, no és un autonumèric.
  - As_Nom: nom de l'assignatura.
  - Crèdits: nombre de crèdits que té l'assignatura.
  - Tipus: obligatòria o optativa.
  - Curs: indica el semestre que s'ha de cursar l'assignatura (1A,1B,2A,2B,3A,3B, 4A i 4B).

- **Carrera:** taula amb les carreres que imparteix l'escola, té tres atributs:
  - Ca_ID: identificador de la carrera, ve donat pel centre, no és un autonumèric.
  - Ca_Nom: nom de la carrera.
  - Any_pla: indica l'any del pla.
  - Previsio_matricules: previsió de nous matriculats en el semestre de tardor.

  - Vigent: indica si és l'actual pla d'estudis. Un 1 indica que si, 0 no.

- **Oferta:** taula on es registra l'oferta d'assignatures de l'actual semestre i el vinent, té quatre atributs:
  - Of_ID: identificador autonumèric.
  - Assignatura: assignatura que s'imparteix.
  - Professor: professor responsable que imparteix l'assignatura. Hi ha assignatures que poden tenir dos o més professors, però les dades que tenim només ens permeten saber el responsable de l'assignatura, per això només es mostrarà un únic professor.
  - Semestre: indica el semestre del qual és l'oferta de l'assignatura.
- **Històric_assignatures:** taula on es guarden les dades necessàries per calcular el % d'aprovats que té una assignatura en concret. Això serà útil perquè a l'hora de saber en quines assignatures es matricularan els alumnes, encara no es sap si han aprovat o no, per tan s'haurà de calcular quin és el % d'aprovats de mitjana per assignar quin alumnes repetiran l'assignatura (els que tinguin pitjor nota mitjana i no l'hagin repetit ja). Aquesta taula té sis atributs:
  - Ha_ID, assignatura, professor i semestre: la mateix explicació que en la taula anterior.
  - Num_matriculats: nombre de matriculats a l'assignatura.
  - Aprovats: % d'alumnes que han aprovat l'assignatura.

- **Semestre:** taula on es registren els diferents semestres en què s'han impartit classes des de que hi ha dades, té quatre atributs:
  - Se_ID: identificador autonumèric.
  - Se_Nom: indica si el semestre és de primavera o tardor.
  - Any: indica l'any.
  - Actual: enter que indica si és el semestre que s'està cursant actualment. Pot prendre dos valors (1 és el semestre actual, 0 no ho és).

- **Matricula:** taula on es guarda les assignatures que està realitzant un alumne o que ja ha realitzat i una vegada calculat el pressupost, també es registren les assignatures que probablement agafarà el semestre vinent. Té diferents atributs:
  - Mat_ID: identificador de l'assignatura.
  - Alumne: identificador de l'alumne.
  - Oferta: identificador de l'oferta d'una assignatura.
  - Assignatura: identificador de l'assignatura.
  - Nota: nota de l'assignatura ja cursada, si l'està realitzant és un -1, si està convalidada és un -2.

- **AsCa:** contracció d'assignatures i carreres. Una assignatura pot ser impartida en més d'una carrera, per això es necessita aquesta taula. Té tres atributs:
  - ID: identificador autonumèric.
  - Assignatura: codi de l'assignatura.
  - Carrera: codi de la carrera.

- **AlCa:** contracció d'alumne i carrera. Un alumne pot està realitzant una (el més freqüent) o més carreres, per això la necessitat d'aquesta taula. Té dos atributs:
  - Alumne: identificador de l'alumne.
  - Carrera: identificador de la carrera.

- **Preu_credit:** taula on es guarden els diferents preus d'un crèdit d'una carrera. Els preus dels crèdits varien cada any, canviant-se al semestre de tardor.
  - Pr_ID: identificador autonumèric.
  - Carrera: identificador de la carrera.
  - Semestre: identificador del semestre.
  - Preu: preu d'un crèdit.
  - Recarrec_1: percentatge de recàrrec al repetir per primera vegada una assignatura.

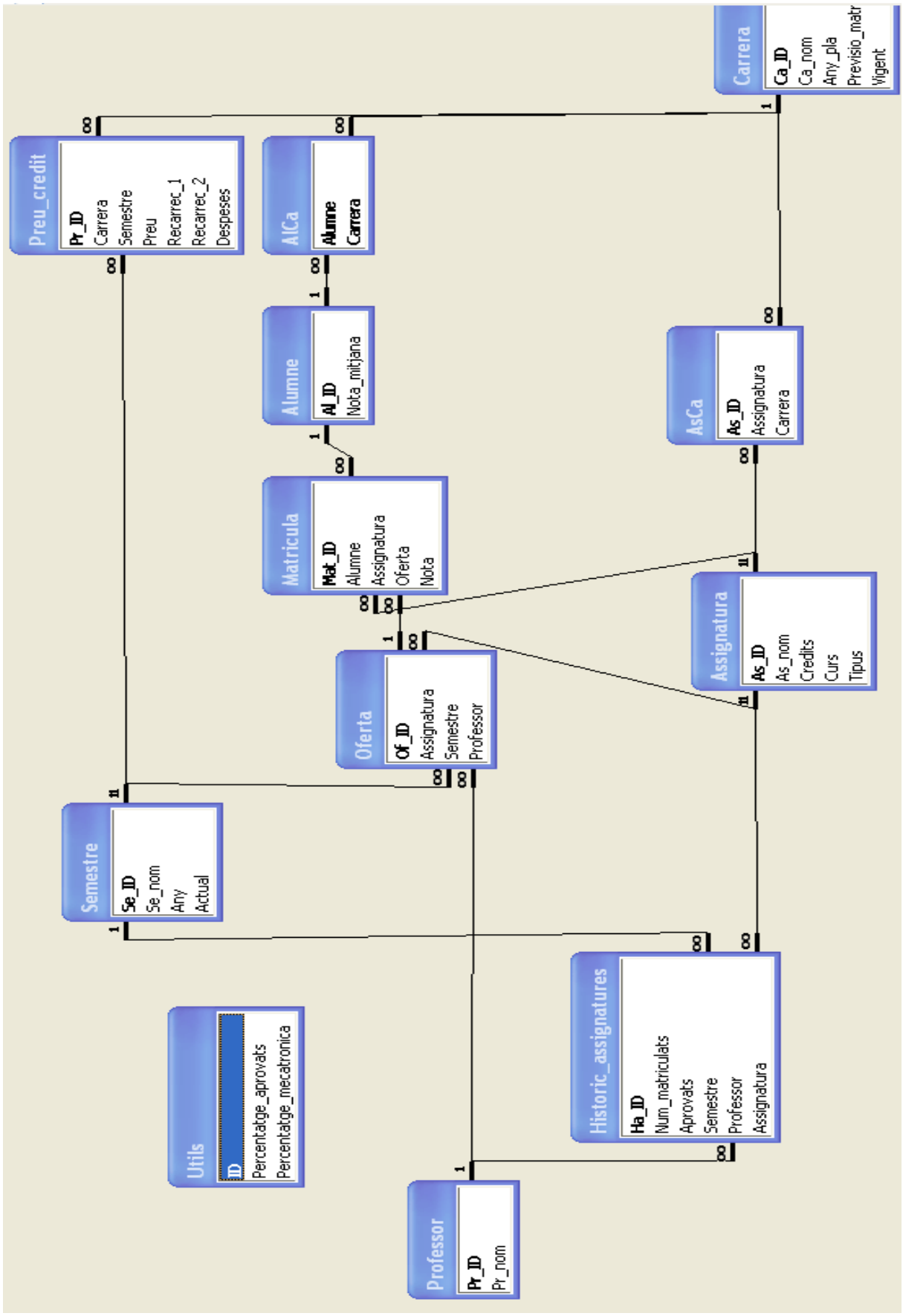
  - Recarrec_2: percentatge de recàrrec al repetir per segona vegada una assignatura.
  - Despeses: cost de realitzar la matrícula, es podrien incloure altres despeses.

- **Utils:** taula on es guarden configuracions del programa.
  - Percentatge_aprovats: les carreres que es fan per primera vegada no és possible saber quan alumnes aproven, per tan es demana que el gestor indiqui quina quantitat acostuma a passar de mitjana.
  - Percentatge_electrònica: a les carreres d'electrònica i mecànica el primer cicle és comú i després es divideix en les seves respectives. No és possible saber quina quantitat de gent farà una o l'altre ja que no hi ha dades anteriors i no es podria fer la previsió. Per tan el gestor del programa haurà de triar manualment el percentatge i en algun lloc s'ha de guardar.

Altres coses a tenir en compte en el model:

- La taula Matrícula està dissenyada per enregistrar les matrícules de l'actual semestre i la previsió del següent. No està dissenyada per ser un històric i per tan, quan es canvia el semestre actual (es passa al següent semestre a gestió de semestres) totes les dades de matrícula són borrades.
- Els alumnes que venen de mòduls tenen certes assignatures automàticament aprovades amb un 5 i només han de pagar un 25% del seu valor. Aquest cost es paga la primera vegada que es matricula.
- Els nous alumnes no tindran un DNI, per tan se'ls assignarà el següent format: 0000000ZZ, 0000001ZZ, etc...

Imatge de com les taules estan relacionades entre si


Carrera
Ca_ID
Ca_nom
Any_pla
Previsio_matr
Vigent
Preu_credit
Pr_ID
Carrera
Semestre
Preu
Recarrec_1
Recarrec_2
Despeses
AlCa
Alumne
Carrera
Alumne
Al_ID
Nota_mitjana
AsCa
As_ID
Assignatura
Carrera
Matricula
Mat_ID
Alumne
Assignatura
Oferta
Nota
Assignatura
As_ID
As_nom
Credits
Curs
Tipus
Oferta
Of_ID
Assignatura
Semestre
Professor
Semestre
Se_ID
Se_nom
Any
Actual
Historic_assignatures
Ha_ID
Num_matriculats
Aprovats
Semestre
Professor
Assignatura
Utils
ID
Percentatge_aprovats
Percentatge_mecatronica
Professor
Pr_ID
Pr_nom

4.7 Ús de patrons

4.7.1 Patró Controlador

Aquest patró està implementat només per la classe del mateix nom a la capa aplicació, és un patró que serveix exclusivament com a intermediari entre una interfície i la resta de classes del projecte. La seva funció és rebre la dades introduïdes per l'usuari de la capa presentació i cridar les funcions o procediments de les capes inferiors (o l'invers també).

L'ús d'aquest patró augmenta la reutilització del codi i se n'obté un major control d'ell (probablement sorgeix d'aquí el seu nom).

4.7.2 Patró Singletó

Aquest patró està dissenyat perquè només es pugui crear una única instància d'un objecte i proporcionar accés, a través d'un mètode, a la resta de l'aplicació.

El patró singletó s'implementa mitjançant un mètode en una classe que cridi el constructor de la classe. Si encara no existeix una instància de la classe, es crea i es retorna la instancia. Si ja existeix, simplement es retorna. També per seguretat, convé posar el constructor de la classe com a privat. Així s'assegura la classe de no se re-instanciada múltiples vegades.

En aquesta aplicació existeixen diferents patrons singletó, connexió (ja que és un recurs únic que només pot ser instanciat una vegada) i la façana.

4.7.3 Patró Façana

Aquest patró és una simple interfície entre dues capes, per tal d'ocultar una capa , la complexitat de l'altre. A través de com un indica el seu nom, una façana.

La funció principal del patró façana és ocultar tot el possible, el conjunt de classes que formen el paquet, de manera que només existeix un punt d'unió entre les dues capes afectades.

L'avantatge que dóna l'ús d'aquest patró és l'augment de la flexibilitat de l'aplicació. Ja que es poden fer canvis a les classes, sense haver de fer cap mena de modificació a l'altre capa .

L'única classe que utilitza aquest patró és la classe façana. La seva missió és ocultar de tapar les classes de persistència i domini. A més de ser l'única classe que comunica la capa superior (domini) amb l'inferior (persistència).

4.7.4 Baix acoblament i cohesió

L'acoblament és el grau de dependència que té un element envers un altre. Per tan, una aplicació té baix acoblament quan els seus elements no depenent gaire d'altres elements del mateix aplicatiu.

La cohesió és la manera en què s'agrupen els elements del software en un element major (les classes en capes per exemple), quan més ben organitzats estan els elements, més alta és la cohesió.

Al llarg de tot el software, s'ha intentat seguir aquestes dues fàcils regles a simple vista tot i que complicades a dur-les a terme.

4.7.5 Patró MVC

Aquest patró separa en tres components diferents: les dades de l'aplicació, la interfície gràfica que veu l'usuari i la lògica de control. La seva utilització és molt important, ja que ajuda amb dos conceptes anteriorment esmenats com són el baix acoblament i l'alta cohesió. També és útil pel fàcil manteniment en el futur de l'aplicació al separar els diferents processos segons els criteris que es considerin adients.

Com a comentari personal, l'aplicació d'aquest patró ha permès que es canviés el SGBBDD (anteriorment s'ha comentat que s'ha substituït l'Access pel MySQL degut a la seva potència superior) i només s'ha hagut de modificar la classe connexió de dins persistència.

5. Planificacions de tasques

A continuació el full de ruta inicial amb les següent consideracions:

- La jornada laboral serà de dilluns a divendres durant 4 hores diàries.
- No s'han tingut en compte els festius ni Setmana Santa.
- La documentació, tot i ser una tasca col·locada al final amb nomes 45 hores, és un procés continuu que s'anirà realitzant al llarg d'aquest mesos de projecte i prendrà hores de treball sobrants d'altres tasques.
- Aquesta planificació s'ha realitzat amb la metodologia UP (Procés Unificat) i en cadascuna de les diferents fases s'han realitzat les tasques d'anàlisi, disseny, implementació, proves i documentació.

Tasca	Hores	Inici	Fi
- Anàlisis		15/02/11	28/02/11
Entrevista	4	15/02/11	15/02/11
Anàlisi de requeriments	12	16/02/11	18/02/11
Definició de la tecnologia i requeriments	12	21/02/11	23/02/11
Definició dels casos d'ús	12	24/02/11	28/02/11
- Disseny		28/02/11	01/04/11
Disseny de la maqueta	60	28/02/11	18/03/11
Disseny de la BBDD	40	21/03/11	01/04/11
- Implementació	180	04/04/11	03/06/11
- Proves	20	06/06/11	10/06/11
- Documentació	45	13/06/11	23/06/11
Total	385		

6. Anàlisi econòmic

El cost del software s'ha realitzat segons la previsió inicial, ja que a l'hora d'entregar un pressupost no hi hauria cap altre manera de fer-ho.

A poques hores de finalitzar la memòria i en conseqüència el PFC, comptabilitzant les hores realitzades realment, la desviació no ha estat gaire gran (402 hores envers les 385 previstes inicialment). En canvi, les hores previstes per a cada apartat han variat substancialment, però això no afecta al pressupost per res.

A l'hora de calcular el pressupost s'han tingut en compte les següents consideracions:

- El preu d'analista i programador és el mateix ja que la persona encarregada a realitzar el projecte ha estat la mateix (servidor).
- Degut a que la inversió ha estat nul·la (per realitzar aquest projecte nomes s' ha requerit un PC) no hi ha despeses de materials ni amortitzacions.
- El preu per hora s'ha obtingut a partir d'el salari anual d'un programador júnior, dividint-lo per 14 pagues i tornant-lo a dividir per 40 hores setmanals s'obté el següent:

16000 / 14 / 40 = 28,57 €

Finalment s'obté el cost del projecte multiplicant pel nombre d'hores dedicades:

10.999,45 €

Ja que aquest projecte serà donat per servidor a la universitat sense rebre cap mena de retribució econòmica a canvi, agrairia que es posés el meu nom a alguna part comuna de l'escola com per exemple Biblioteca Narcís Mir. Tot i que em conformaria amb una estàtua de bronze al mig de la plaça, homenatjant a la meva persona i així els futurs alumnes poguessin rendir-me culte i pleitesia.

* Nota sobre l'estil: M'he permès la llicència de saltar-me l'estil impersonal en aquest apartat, per transmetre una opinió i un desig. Tot i que he intentat escriure-ho parlant de mi en tercera persona, m'ha provocat tal xoc, que ho he hagut d'esborrar. No m'agradaria haver d'afegir despersonalització a la meva llarga i feixuga llista de tares mentals.

7. Ampliacions futures

Aquesta aplicació compleix amb totes les tasques que se li van encomanar i per tan, dels requeriments inicials no es pot realitzar cap ampliació.

Ja que l'apartat de funcions està cobert, quedaria la part d'optimitzar i millorar el que està fent actualment. Tot i realitzar bones previsions seria interessant que una vegada s'han obtingut les matrícules del nou semestre, el software comparés amb la previsió que va realitzar i a partir d'aquí s'ajustés els paràmetres ell sol.

Aquí s'està entrant en terreny d'intel·ligència artificial i nous llenguatges de programació com el PROLOG. El temps requerit per aprendre la base del llenguatge i aplicar-lo per poder realitzar el anteriorment esmenat permetria realitzar un nou PFC. Això ho deixo per les generacions futures.

8. Conclusions

8.1 Conclusions sobre el projecte

El projecte ha arribat correctament a terme i ha complert satisfactòriament les expectatives que s'havien establert a l'inici durant els requeriments.

Destacar la sorpresa al trobar un SGBBDD com l'Access totalment sobrepassat per una aplicació com aquesta. Cal dir també, que fins ara un servidor no havia treballat en volums de dades tan importants. Aquest succés va comportar la substitució d'aquest SGBBD per un altre de més potent i a sobre obert com el MySQL. Gràcies a la programació modular, aquest canvi ha estat només una anècdota i una manera de veure les carències de l'Access.

8.2 Conclusions personals

Del treball en línies generals puc dir que n'estic satisfet, que no orgullós. Crec que la meva opinió sobre aquesta carrera i aquest món (la informàtica en general) ha quedat plasmada vàries vegades a base de sarcasmes al llarg del text.

Veig el meu futur lluny dels ordinadors i molt més de les pantalles plenes de línies de codi. La realització d'aquest projecte només ha estat la enèsima confirmació d'una cosa que sabia, una equivocació que ha costat cinc anys.

Ara que estic acabant d'escriure les últimes línies d'aquest treball, no puc deixar de mirar enrere i tot i haver de fer les conclusions personals d'aquest treball, les faré de tota la carrera. Perquè de fet, aquest treball no deixa de ser la seva culminació.

Crec que porto tota la memòria fent el balanç negatiu, seria just que comencés a fer el positiu, encara que aviso que serà curt. Tot i avorrir les matèries i els conceptes donats, darrere d'això sempre hi ha una base, una metodologia i un pensament. Així que tot i el què he dit, a dins meu sempre hi haurà un enginyer.

Cinc anys són molts, poder masses i tot. Això permet conèixer grans persones que t'han acompanyat durant aquest llarg trajecte fent-lo més amè. Una part d'aquestes persones són els

companys: Xavi, Xesc, Dani, Cabe i Felipe gràcies per compartir aquesta etapa de la vida. Ara en començarà una de nova, el temps dirà si els nostres camins es tornen a creuar.

L'altre part de persones, alguns professors com l'Eduard, gràcies per portar-me el projecte i donar-me aquesta última oportunitat, espero que el resultat no t'hagi defraudat. També vull disculpar-me amb en Xavier Font: per portar-me un PFC que al final no es va a dur a terme. Aquell semestre vaig tirar la tovallola fastiguejat i vaig abandonar-ho tot. He necessitat un any per tenir la suficient determinació per acabar una cosa que no havia ni d'haver-la començat. Finalment vull anomenar a la Núria, més com a professora com a persona. Pel seu esforç i dedicació a l'hora de realitzar les classes i el tracte proper i amigable que ha tingut amb tots els seus alumnes. Només vull dir-te que gràcies a tu m'he adonat de l'error que he comès. Una vegada realitzades les teves dues assignatures i una d'optativa te n'adones de que amb les que més he gaudit i he trobat més interessants han estat les teves i dius: d'acord, quan vaig triar entre l'enginyeria i econòmiques crec que em vaig equivocar...

Sé que he tornat a trencar la normativa i he acabat escrivint uns agraïments a un lloc que no tocaven. Demano disculpes, ha estat totalment espontani. L'únic que puc fer, és assegurar que això no tornarà a passar. No perquè no ho tornaria a fer, sinó perquè això és el penúltim capítol i a la bibliografia quedaria massa lleig. Sé que m'he pres masses llicències, però jo sóc jo i les meves circumstàncies i aquest és el meu projecte.

9. Referències

Llibreria jexcelapi

http://jexcelapi.sourceforge.net/

http://www.andykhan.com/jexcelapi/tutorial.html

SQL

http://www.w3schools.com/sql/default.asp